

In the Claims:

Please amend claims 1, 2, 7, 9, 15, 19, 21, 22 and 26, cancel claims 9-14, and add claims 31-33. All claims in the application are reproduced below.

1. (Currently Amended) An interactive tool for manipulating a plurality of deployment descriptors, comprising:

a plurality of applications deployed on a web server, wherein each one of the plurality of applications is associated with one of the plurality of deployment descriptors that describes deployment and configuration information of the application on the web server;

~~a first user interface capable of rendering a hierarchical representation of the plurality of deployment descriptors, wherein a component of the representation can be selected by a user;~~

~~a second user interface capable of rendering a user editable representation of the selected component;~~ and

a builder component capable of

~~organizing deployment information from all of the plurality of deployment descriptors into a logical hierarchy of resources associated with the web server; and~~

~~maintaining a data structure to represent a state of the logical hierarchy of resources associated with the plurality of applications deployed on the web server at a given time~~

creating a master tree data structure based on the present state of all deployment descriptor files, wherein the master tree data structure represents a state of the logical hierarchy of resources associated with the plurality of applications at a given time;

creating a separate tree data structure that represents deployment descriptor information based on the current state of source files in an application's project directory, wherein the separate tree data structure represents a state of the logical hierarchy of resources associated with the application, wherein the application is one of the plurality

of applications deployed on the web server;
comparing the master tree data structure with the separate tree data structure; and
refreshing the master tree data structure based on the separate tree data structure,
if the master tree data structure is different from the separate tree data structure.

2. (Currently Amended) The interactive tool of claim 1, further comprising:
a ~~third~~ user interface capable of rendering an error message.
3. (Original) The interactive tool of claim 2 wherein:
user selection of the error message can cause the second user interface to render a user-editable representation of the at least one deployment descriptor component that is in error.
4. (Original) The interactive tool of claim 1, further comprising:
a parser capable of generating a representation of the at least one deployment descriptor;
a generator capable of creating the at least one deployment descriptor; and
a validator capable of validating the at least one deployment descriptor.
5. (Original) The interactive tool of claim 4 wherein:
the validator is capable of generating an error when it encounters a syntactic or semantic fault in the at least one deployment descriptor.
6. (Previously Presented) The interactive tool of claim 1, wherein:
the builder component is further capable of automatically updating the at least one deployment descriptor to reflect one or more changes in at least one source code file.
7. (Currently Amended) The interactive tool of claim 1 wherein:

the hierarchical representation can include information pertaining to an archive file at least one of: a Java™ archive (JAR), a Web Archive (WAR), an Enterprise Archive (EAR), and a Java™ Connector Architecture Component (RAR).

8. (Original) The interactive tool of claim 1 wherein:

the at least one deployment descriptor can be expressed as an Extensible Markup Language document.

9-14. (Canceled).

15. (Currently Amended) A method for providing an interactive tool for manipulating a plurality of deployment descriptors, comprising:

deploying a plurality of applications on a web server, wherein each one of the plurality of applications is associated with one of the plurality of deployment descriptors that describes deployment and configuration information of the application on the web server;

~~providing a first user interface capable of rendering a hierarchical representation of the plurality of deployment descriptors, wherein a component of the representation can be selected by a user;~~

~~providing a second user interface capable of rendering a user editable representation of the selected component;~~

~~providing a third user interface capable of rendering an error message;~~

~~organizing deployment information from all of the plurality of deployment descriptors into a logical hierarchy of resources associated with the web server; and~~

~~maintaining a data structure to represent a state of the logical hierarchy of resources associated with the plurality of applications deployed on the web server at a given time~~

creating a master tree data structure based on the present state of all deployment

descriptor files, wherein the master tree data structure represents a state of the logical hierarchy of resources associated with the plurality of applications at a given time;

creating a separate tree data structure that represents deployment descriptor information based on the current state of source files in an application's project directory, wherein the separate tree data structure represents a state of the logical hierarchy of resources associated with the application, wherein the application is one of the plurality of applications deployed on the web server;

comparing the master tree data structure with the separate tree data structure; and
refreshing the master tree data structure based on the separate tree data structure, if the master tree data structure is different from the separate tree data structure; and

~~wherein user selection of the error message in the third user interface can cause the second user interface to render a user editable representation of the at least one deployment descriptor component that is in error.~~

16. (Original) The method of claim 15, further comprising:

providing a parser capable of generating a representation of the at least one deployment descriptor;

providing a generator capable of creating the at least one deployment descriptor; and
providing a validator capable of validating the at least one deployment descriptor.

17. (Original) The method of claim 16 wherein:

the validator is capable of generating an error when it encounters a syntactic or semantic fault in the at least one deployment descriptor.

18. (Original) The method of claim 15, further comprising:

providing a builder component capable of automatically updating the at least one deployment descriptor to reflect one or more changes in at least one source code file.

19. (Currently Amended) The method of claim 15 wherein:

the hierarchical representation can include information pertaining to an archive file at least one of: a Java™ archive (JAR), a Web Archive (WAR), an Enterprise Archive (EAR), and a Java™ Connector Architecture Component (RAR).

20. (Original) The method of claim 15 wherein:

the at least one deployment descriptor can be expressed as an Extensible Markup Language document.

21. (Currently Amended) A machine readable medium having instructions stored thereon that when executed by a processor cause a system to:

~~deploying deploy~~ a plurality of applications on a web server, wherein each one of the plurality of applications is associated with one of a plurality of deployment descriptors that describes deployment and configuration information of the application on the web server;

~~provide a first user interface capable of rendering a hierarchical representation of the plurality of deployment descriptors, wherein a component of the representation can be selected by a user;~~

~~provide a second user interface capable of rendering a user editable representation of the selected component;~~

~~provide a third user interface capable of rendering an error message; and~~

~~organize deployment information from all of the plurality of deployment descriptors into a logical hierarchy of resources associated with the web server; and~~

~~maintain a data structure to represent a state of the logical hierarchy of resources associated with the plurality of applications deployed on the web server at a given time~~

create a master tree data structure based on the present state of all deployment descriptor files, wherein the master tree data structure represents a state of the logical hierarchy of resources associated with the plurality of applications at a given time;

create a separate tree data structure that represents deployment descriptor information based on the current state of source files in an application's project directory, wherein the separate tree data structure represents a state of the logical hierarchy of resources associated with the application, wherein the application is one of the plurality of applications deployed on the web server;

compare the master tree data structure with the separate tree data structure; and
refresh the master tree data structure based on the separate tree data structure, if the master tree data structure is different from the separate tree data structure.

22. (Currently Amended) The machine readable medium of claim 21, further comprising instructions that when executed cause the system to ~~wherein~~:

provide a user interface capable of rendering an error message, wherein user selection of the error message in the third user interface can cause the second user interface to render a user-editable representation of the at least one deployment descriptor component that is in error.

23. (Original) The machine readable medium of claim 21, further comprising instructions that when executed cause the system to:

provide a parser capable of generating a representation of the at least one deployment descriptor;

provide a generator capable of creating the at least one deployment descriptor; and

provide a validator capable of validating the at least one deployment descriptor.

24. (Original) The machine readable medium of claim 23 wherein:
the validator is capable of generating an error when it encounters a syntactic or semantic fault in the at least one deployment descriptor.

25. (Original) The machine readable medium of claim 21, further comprising instructions that when executed cause the system to:
provide a builder component capable of automatically updating the at least one deployment descriptor to reflect one or more changes in at least one source code file.

26. (Currently Amended) The machine readable medium of claim 21 wherein:
the hierarchical representation can include information pertaining to an archive file at least one of: a Java™ archive (JAR), a Web Archive (WAR), an Enterprise Archive (EAR), and a Java™ Connector Architecture Component (RAR).

27. (Original) The machine readable medium of claim 21 wherein:
the at least one deployment descriptor can be expressed as an Extensible Markup Language document.

28. (Previously Presented) The interactive tool of claim 1, wherein:
the interactive tool is capable of automatically repairing a first deployment descriptor of the at least one deployment descriptors if the first deployment descriptor is defective.

29. (Currently Amended) The interactive tool of claim 1, wherein:
the builder component is further capable of create creating a tree data structure

Application No.: 10/772,613
Reply to Office Action dated: April 1, 2009
Reply dated: July 1, 2009

that embodies hierarchical relationships of nested XML statements.

30. (Canceled).

31. (New) The interactive tool of claim 1, wherein:

the builder component is further capable of keeping a module in the master tree data structure to allow applications other than a current application to use the module, even after the module is removed from the current application.

32. (New) The interactive tool of claim 1, further comprising:

a first user interface capable of rendering a hierarchical representation of the plurality of deployment descriptors, wherein a component of the representation can be selected by a user; and

a second user interface capable of rendering a user-editable representation of the selected component.

33. (New) The interactive tool of claim 1, wherein:

the builder component is further capable of generating a new deployment descriptor for the application from the refreshed master tree data structure.